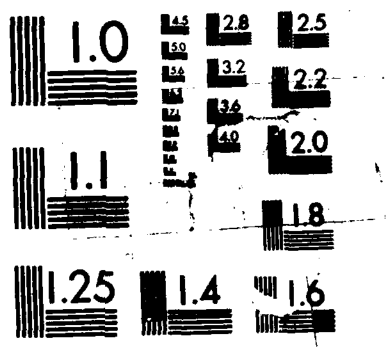
AD-A187 243 RESEARCH ON INTERACTIVE PROTOCOLS(U) LOCKHEED ELECTRONICS CO INC PLAINFIELD NJ HUMAN MACHINE SYSTEMS GROUP 14 DEC 84 DAAK11-84-M-0036 1/1

UNCLASSIFIED F/G 23/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(2)

RESEARCH ON INTERACTIVE PROTOCOLS
Contract: DAAK-11-84-M-0036
FINAL REPORT

Human Engineering Laboratory
Aviation and Air Defense Directorate
Aberdeen Proving Ground, Maryland 21005
14 December 1984

Human Machine Systems Group
System Analysis Department
Lockheed Electronics Company, Inc.
Plainfield, New Jersey 07061

CONTENTS

1.0 Introduction

The purpose of this contract was to "examine the soldier-machine interface of automated battlefield intelligence/C^2_n systems including optimization of operator-computer protocols." The work statement tasks to support this effort were:

1) storyboard the application demonstration based on the scenario developed by HEL/AAAD. This shall include the selection of one or two operator tasks and shall use the specific data presented in the scenario;

2) analyze the operator tasks to determine the possible alternative protocols which could be used to execute them. Ideally, the chosen tasks are ones for which more than one protocol can be used;

3) design data display formats, corresponding to different protocol types, for the operator tasks;

4) write the graphics routines and author the GHMI system to provide for the demonstration of the tasks. This shall include being able to demonstrate the alternative protocols for the same task;

5) integrate, test and debug the demonstration;

6) host a one-day meeting with HEL/AAAD; and

7) document the effort, providing descriptions of the operator tasks, the alternative protocols used, pictures (or photographs), and recommendations for future studies.

The first five tasks are discussed in Sections 1.0 through 5.0 of this report. Specific research questions, raised during the demonstration development effort, are discussed in Sections 3.0 and 4.0; general considerations for future research are described in Section 5.0. A summary of the one-day meeting held at Lockheed on 7 December 1984 appears in Section 7.1 (Appendix 1).

2.0 Data Analyses, Task Selection, Demonstration Specification

To support development of a protocol demonstration, the HEL Aviation and Air Defense Directorate (HEL/AAAD) provided a one-half hour "slice" of a Short Range Air Defense System Command and Control (SHORADS C^2) scenario. The data provided by HEL/AAAD consisted of friendly and enemy air tracks, the deployment of an ADA Heavy Division SHORAD battalion (reduced by 20% due to battle attrition) in the area of interest, and chronologically-ordered message traffic to and from the Air Battle Management Operations Center (ABMOC). The type and quantity of the ABMOC messages were developed in accordance with the

benchmark requirements specified in MIS 34585, "SHORAD C^2 System Specification" (10 October 1983).

The initial task on this contract was the analysis of these data to understand the battlefield situation (deployment of friendly fire units, sensors and other assets, unit organization, battlefield geometry), and the message flow in and out of the ABMOC during the half-hour period. Due to the limited funding available for execution of this contract, it was decided that it would not be possible to include air track data and track-related messages in the demonstration.

A picture of the battlefield situation specified by the scenario data is presented in Figure 1. The breakdown of TO-ABMOC messages (messages sent from other SHORADS C^2 nodes to the AMBOC) and FROM-ABMOC messages (messages composed at the ABMOC and addressed to other nodes) during the half-hour period, and a few examples of each type, are shown in Tables 1 and 2, respectively.

It became obvious after plotting the battlefield situation that a means for selectively displaying different battlefield features would be important. This would allow the user to declutter the display of information so that only those data that were relevant at some particular moment would be presented. Therefore, a method for decluttering the situation display was designed and demonstrated. As we were developing this part of the demonstration, it became apparent that details of the situation display, such as unit designations, would not be discriminable when the full battlefield (a 70km-X-70km area) was presented, and, if shown on the full map, would only add clutter without adding any information. This led us to developing a method for "blowing up" a smaller section of the map where detailed information would be provided. Details of this part of the demonstration are described in Section 3.0.

Based on the data that were provided (situation display and messages), prior discussions with former AD personnel, and previous experience with the SHORADS C^2 operational concept, it seemed that the situation display was the most important piece of information to ABMOC personnel and that a user would be assisted if his message processing tasks could be performed without requiring that he look someplace other than at the display (such as at a keyboard). We also imagined that the user "thinks" primarily in spatio-temporal terms: he is most often concerned with elements and events occurring at some location at some point in time. All scenario messages provided to us as data were represented in alphanumeric form. If a user actually had to read and compose alphanumeric messages, we anticipated that he would experience the following problems:

- conversion of location information (e.g., FH368243) from alphanumerics into an x-y coordinate on his display, and vice versa, would be a cognitively difficult and demanding task,

- since he could not be looking simultaneously at two places, reading or composing a message would require that he take his eyes away from the situation display, and

- the user's "mental workload" would be increased by having to mentally correlate a particular (graphically displayed) unit with the message it sent, or was to receive.

These considerations led us to focus our protocol design efforts on methods which would represent message information in graphical forms and would minimize both the distance and the frequency with which the user would have to move his eyes from the the situation display.

Since it was not possible to demonstrate processing for all message types, the subset selected and their implementation order were determined by the frequency of message type, the relative importance of the message type, and the opportunity provided by the message type to demonstrate certain interactive protocol features that we consider to be important. (Appendix 2 contains R. E. Knox and R. Brandau "Interactive protocols for generalized human-machine interface;" it presents a general discussion of protocol design issues.) In the time provided, it was possible to demonstrate all TO-ABMOC messages for which an implementation order is shown in Table 1 (Message ID #1, 4, 12, 20, 29); of the FROM-ABMOC messages, only Movement Order composition was demonstrated.

No alternative protocols were demonstrated for the same message type, though different protocol types (e.g., "picking", "locating") are used for different tasks. A discussion of possible alternative protocols and analytic arguments for why they would be less desirable (i.e., human performance would suffer) are presented in Section 4.0.

3.0 Description of Demonstration

The hardware and software facilities used for this demonstration were:

- a DEC PDP-11/70 (used for the simulation of message traffic described in section 3.2); simulation software is written in Pascal 2 (Oregon Software, Inc.) and runs under the RSX-11M-PLUS (DEC) operating system;

- a Codata 68000-based system (used for control of the "ABMOC" console); software developed on this system runs under the UNIX (Bell Laboratories) operating system; some application processing code is written in the C language; all interactive graphics software is written in Easel (Interactive Images Incorporated);

Figure 1
Scenario Battlefield Situation.

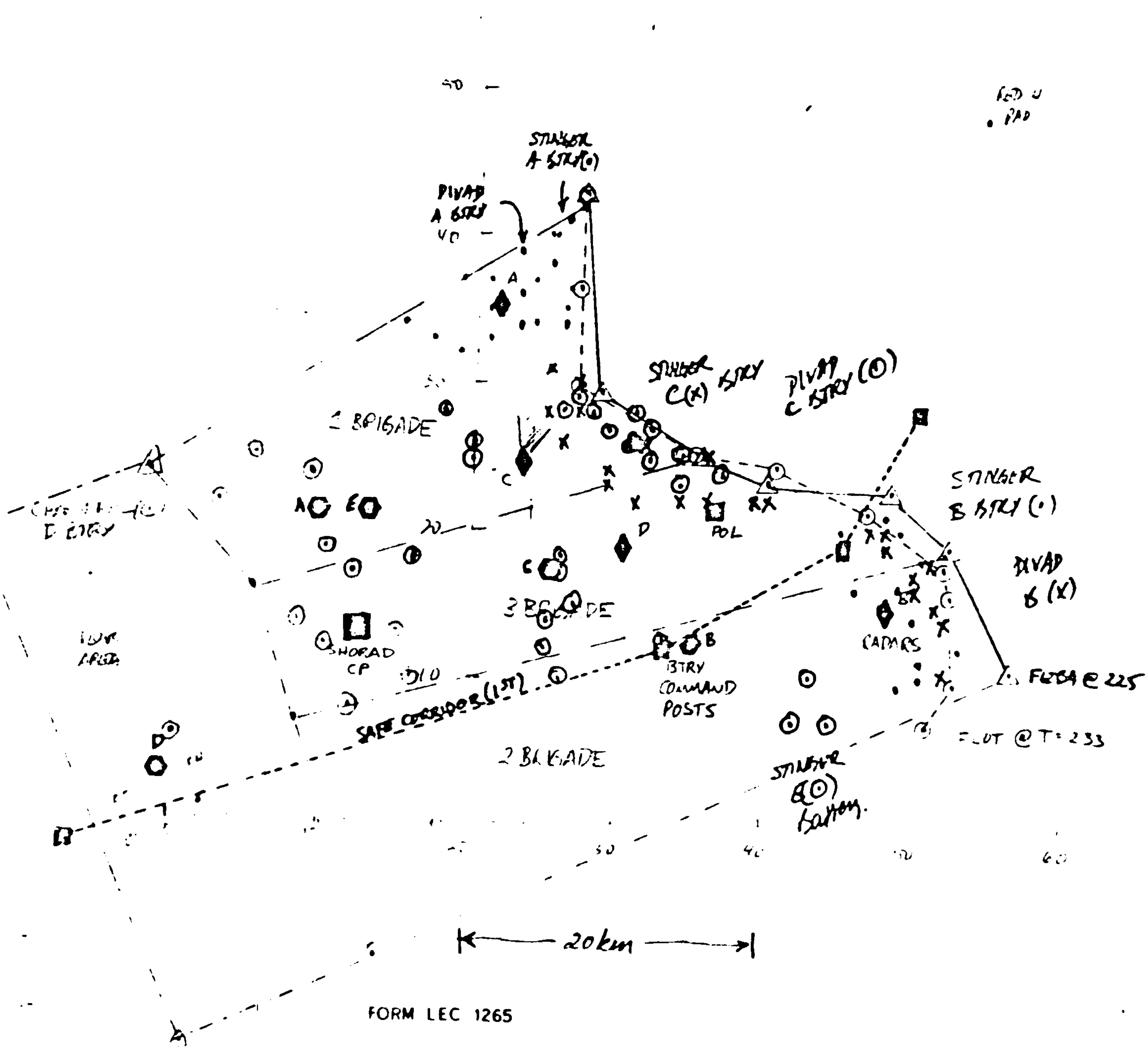


Table 1
TO-ABMOC Messages

ID #	TYPE OF MESSAGE	Count (#)	IMPLEMENTED	Warnings VIS	Warnings AUD	ACK Reqt.?
1	(Air Defense Warning):	1	4th	X	X	X
4	(Battlefield Geometry):	1	3rd	-	-	-
5	(Data Link Reference Point):	2		X	-	-
9	(Enemy Activity Report)	1		X	-	-
12	(Kill Report):	8	2nd	X	-	-
19	(Pointer):	1		X	-	X
20	(Reporting Unit Location):	44	1st	-	-	-
23	(Supply & Equip. Status Rept.):	3		X	-	-
24	(Track Management):	6	NO	-	-	-
27	(Unit Operational Report):	3		X	-	-
28	(Warning Report):	5		X	-	X
29	(Weapon Control Order):	1	5th	X	-	X

Examples

1045 (Scenario Time)

1. FROM: FAAR A
TO: ABMOC
MESSAGE TYPE: 20 (Reporting Unit Location) (PLRS))
MESSAGE: GRID FH215354 ALT 299

1046

2. FROM: 1/4/A
TO: ABMOC
MESSAGE TYPE: 20 (Reporting Unit Location)
MESSAGE: 1/4/A to FH200332

1046

3. FROM: 2/4/A
TO: ABMOC
MESSAGE TYPE: 12 (Kill Report)
MESSAGE: FW 1 DTG 231046 (F/S)

Table 2
FROM-ABMOC Messages

ID #	TYPE OF MESSAGE	Count (#)	IMPLEMENT
6	(Data Management):	1	2nd
11	(IFF/SIF Effective Code):	1	
13	(Movement Order):	9	1st
21	(Sensor Management):	1	
22	(Summary Unit Status):	1	3rd
24	(Track Management):	9	NO

Examples

FROM: ABMOC
TO: FAAR A
MESSAGE TYPE: 13 (Movement Order)
MESSAGE: FAAR A from FH 215354 to FH 116360 at 231800, GS

FROM: ABMOC
TO: E Btry
MESSAGE TYPE: 13 (Movement Order)
MESSAGE: Move weapons to FH 538082

FROM: ABMOC
TO: A Btry
MESSAGE TYPE: 6 (Data Management)
MESSAGE: Data Update Request: Supply/Equip Status

- the Generalized Human Machine Interface console; the devices used for this demonstration were the Mitsubishi 1024-X-1024 color display, overlaid by a high resolution Elographics touch screen, and a Prose 2020 Text-to-Speech synthesizer (Telesensory); the Motorola monochrome display was used only to display a "trace" of the incoming alphanumeric messages that were being converted and displayed on the color monitor.

The following sections describe the demonstration and the protocol conventions used for the different tasks that were implemented. Obviously, to control the length of this document, the description is not exhaustive; rather, we will provide more detail where important issues of interactive protocol design arise.

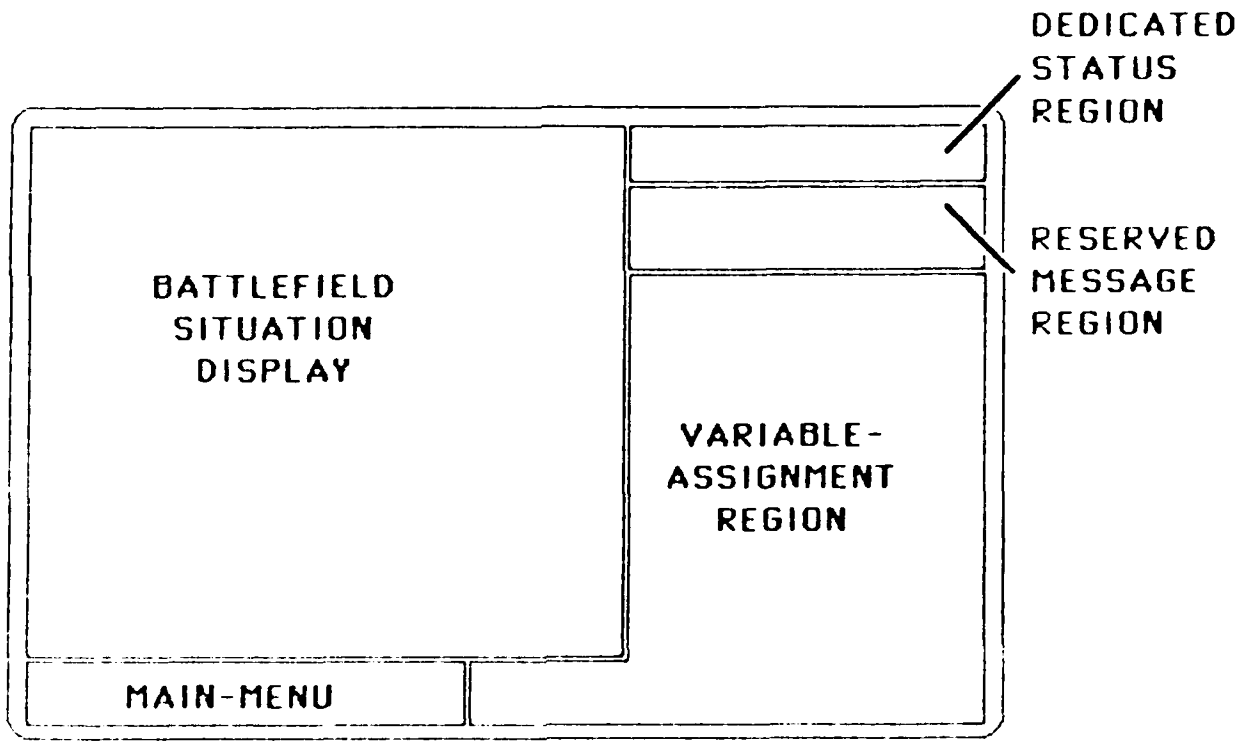
3.1 Situation Display

Figure 2 shows a photograph of the beginning display screen for the demonstration. [All Figures which are photographs are included as Appendix 7.3 of this document]. The dedicated and variable-assignment regions of the software-programmable interactive display screen are defined in Figure 3. The weapons control and air defense warning (ADW) states at the beginning of the half-hour slice (time is 1045) are shown at the top right of the screen; note that redundant coding of the ADW is accomplished through color.

We have adopted the convention that binary-function keys (e.g., on-off, display-no display) are coded by a box surrounding the function label (whether that label is alphanumeric or graphic); functional state changes are accomplished by directly touching the software key on the display screen. The enabled state of the function (e.g., "on," "display") is coded as a change from white lettering on different colored backgrounds (used to code other information, as discussed below) to reverse-video black lettering on a green background. In Figure 2, the main-menu at the bottom of the screen indicates that the "SET-UP-DISPLAY" function is active.

The SET-UP-DISPLAY menu is shown in the variable function area on the right of the screen. For demonstration purposes, we chose a default starting state in which no friendly assets are displayed; this decision, of course, was arbitrary and could be modified to show those elements that are meaningful at system initialization.

Figure 3
Software-programmable interactive display screen layout.



The top section of the menu shows a breakdown of assets by unit type (i.e., DIVAD-"DVAD", STINGER-"STNG", CHAPARRAL-"CHAP", and COMMAND POSTS-"C P") and Task Organization (Brigades-"BDE", Task Force-"TF ELY", and Division Troops-"DIV TRP"); cells within the matrix show the breakdown to the battery level organization. The asset and task organization main levels are differentiated from the battery-level breakdown by using white keys on a blue background; it also provides visual emphasis to the fact that pressing row or column keys will cause all elements within a row or column to be displayed. If the user wished to see all units, he would press the ALL key in the upper-left of the matrix and all units would be displayed as shown in Figure 4. If, at this point, he wished to display only DIVAD units, he has at least two methods of doing so: (1) press the ALL key once again to erase all units, and then press the DVAD key to display only DIVADs, or (2) press the STNG, CHAP, and C P keys to return them to an off state (i.e., erase those units from the display), leaving only the DIVADs displayed. This final state is shown in Figure 5. In terms of the number of key presses required, method (1) is more efficient; on the other hand, if the user wished to see each unit type as it was being removed in the process of displaying only the DIVADs, he would use method (2). The point is that one cannot define efficiency simply in terms of the number of key presses required.

The lower section of this menu allows the user to selectively display ASSETS and Battlefield Geometry (BTFLD GEO) features. The full battlefield situation with all features displayed is shown in Figure 6. Note that the FLOT label appears in the BTFLD GEO submenu, but no box surrounds it. Following our convention, this would imply that the user cannot display the FLOT. In fact, in this scenario, the FLOT is not defined until it arrives as a Battlefield Geometry message at about 1055; we describe in Section 3.2 how the FLOT key is made available to the user when the information is available for display.

The symbols used for representation of battlefield geometry features follow the conventions specified in FM 21-30, "Military Symbols." For unit symbols, we had originally used the symbology defined in Appendix V of MIS 34585 (see Reference [1]). This symbology does not differentiate among the different kinds of fire units; we think such differentiation is necessary. We did not want to color-code the symbols because of the limited number of colors available that can be easily distinguished. Therefore, we adopted a simplified shape-coding scheme: DIVADs were rectangles, STINGERs were triangles, and CHAPARRALs were squares; all friendly fire units were color-coded aqua. The four FAARs were shape-coded as diamonds. A picture of the battlefield situation using this coding is shown in Figure 7. During the demonstration meeting with HEL/AAAD personnel, it was suggested that we use the standard air defense symbology; the battlefield situation using this coding is shown in all preceding photographs other than Figure 7. While our subjective impression is that the simplified, geometric shapes are much easier to discriminate, the use of a standard symbology would be required in an operational setting.

The large situation display only shows unit designation labels for the Battery Command Posts; adding the three- to four-character alphanumeric unit labels to the large display would have added a great deal of clutter. Yet, in an operational setting it will certainly be necessary at times to know specifically who a unit is, or to inspect an area of concern in greater detail. With this in mind, we added the INSPECT function which expands an area of the situation map and displays it in the variable function area. Figure 8 shows the user prompt after the INSPECT function key has been pressed. The user now touches an area on the situation display and an approximately 9 km-by-9 km area, centered on the touch location, appears in the auxiliary map area, as shown in Figures 9a (geometric shape coding) and 9b (standard AD symbology). Note that map symbols are enlarged and unit designation labels are added. It has been suggested that we might have shown the blow-up of the situation display directly on the large map; we chose not to do this as the detailed map would obscure a large section of the current situation; we judged this to be undesirable from an operational point of view.

3.2 TO-ABMOC Message Processing

All TO-ABMOC messages were converted to an on-line data base residing on a PDP-11/70. A small "communications network" simulation program was written; it reads the time at which TO-ABMOC messages are "received" (from the message data base), and sends the set of messages occurring during each one-minute interval to another computer (a Codata 68000 system) where the ABMOC "console" demonstration was implemented.

Five message types were selected for demonstration; these accounted for 55 of the 76 incoming ABMOC messages during the half-hour scenario. The principles guiding the design of message representation were:

- to reduce the amount of alphanumeric data to be read,

- to develop graphic representations of messages that were integrated with the situation display whenever possible, and

- to spatially integrate (i.e., co-locate) new information and user-input requests with existing information.

Implementation of these principles are described in the next five sections.

3.21 Reporting Unit Location

Of the 76 messages received during the half-hour interval, 44 were updates on friendly asset positions. An example of a "reporting unit location" message is shown in Table 3 (for clarity, most battlefield situation features that are irrelevant to this and subsequent examples

of message processing, have been removed from the screen). In representing position changes graphically, we chose to indicate both the old and the new locations (or, in the case of FAAR updates, old and new altitudes); if it is important to know how far and from where a fire unit has moved, both locations would have to be indicated. If both locations are to be observed, they must be displayed for some period of time. Finally, the user must be able to distinguish between the old location and new location indicators; hence, they must be coded in a discriminable fashion.

Table 3
Example of Reporting Unit Location Message

1046 (Scenario Time)

FROM: 2/4/B
TO: ABMOC
MESSAGE TYPE: 20 (Reporting Unit Location)
MESSAGE: 2/4/B to FH473215

Figure 10a shows an example of an old and new location of a STINGER unit; the old location is indicated by the green STINGER symbol and the new location is indicated by the oversized (relative to other symbols) green square. Green was used to highlight the two symbols. It was chosen because of the eye's greater sensitivity to wavelengths in the mid-range of the visible spectrum; yellow was not used because its widespread use to indicate cautionary states. Both symbols remain visible for about five seconds; afterwards, the aqua STINGER symbol is displayed in the new location, as shown in Figure 10b.

3.22 Kill Report

Kill reports specify the unit reporting the kill(s), the type (fixed or rotary wing) and number of aircraft downed, and the date-time group (DTG) of the report; an example of such a message is shown in Table 4.

Table 4
Example of Kill Report Message

1046 (Scenario Time)

FROM: 2/4/A
TO: ABMOC
MESSAGE TYPE: 12 (Kill Report)
MESSAGE: FW 1 DTG 231046 (F/S)

Figure 11 presents an example of a graphics representation of such a message. The STINGER unit (2/4/A) reporting the kill is surrounded by a yellow box. The "report" itself is displayed in the upper right of the situation display: the AD fixed-wing aircraft symbol denotes the type of the downed aircraft, the X above it is the conventional "destroyed" indicator, and the DTG of the report is shown at the bottom of the composite symbol. The yellow color coding was used for distinctiveness, to associate the unit reporting the kill with the report itself (this may, in fact, add information about the approximate location of the kill), and for its implication of a cautionary state.

A kill report remains displayed until a new one is received (see Appendix 1 for a brief discussion of when data should be removed from the display screen). Kill reports would not necessarily always be displayed in this portion of the display (we put it there because it would occlude no other data); rather, we anticipate that "smart" software would "find" the sparsely used portions of a situation display, and would plot such data to minimize the overlay of other relevant situation information. Kill reports, then, would probably be presented at different locations as the scenario progressed. Rather than creating confusion, we expect that such shifting in location would call attention to the arrival of new information. (In fact, one criticism of the demonstration was that it was hard to tell when a new kill report had been received since the data always appeared in the same location.)

3.23 Battlefield Geometry

The one Battlefield Geometry message received during the half hour provides a ten-point definition of the FLOT; the message is shown in Table 5. In the demonstration, this message was converted to the FLOT contour and plotted on the situation map; it is color-coded blue to signify that it borders friendly assets and to differentiate it from the red FEBA. Figure 12 shows this representation.

Table 5
Example of Battlefield Geometry Message

1053 (Scenario Time)

FROM: (D)AME
TO: ABMOC
MESSAGE TYPE: 4 (Battlefield Geometry)
MESSAGE:

REFERENCE NUMBER: ADA4
EFFECTIVE TIME: Upon Receipt
TYPE: FLOT 10 POINTS
LOCATION: From FH270430 to FH270365 to FH270298
to FH350254 to FH407244 to FH468214
to FH520178 to FH522158 to FH527097
to FH510068

At the same time, the FLOT key in the BTFLD GEO submenu of the Set-Up Display function has been enabled and set to the display-state (as indicated by black lettering on a green background); this is also shown in Figure 12. The key is now fully functioning and allows the user to display or erase the FLOT from the situation display as necessary.

3.24 Air Defense Warning

One Air Defense Warning (ADW) is received during the scenario; the message is shown in Table 6. When an ADW is distributed, visual and auditory alerts must be issued to the user; he, in turn, is required to manually (i.e., explicitly) acknowledge that he has received the message.

==
==

Table 6
Example of Air Defense Warning Message

1055 (Scenario Time)

FROM: HIMAD
TO: ALL
MESSAGE TYPE: 1 (Air Defense Warning Report)
MESSAGE: AIR DEFENSE WARNING CONDITION: RED

==
==

Figure 13 shows the change to the ADW indicator in the status region at the upper right of the display screen. The instruction to acknowledge receipt of the ADW is presented in the reserved message region just below the status region. The auditory alert, presented simultaneously with the ADW change and annunciated by the text-to-speech voice synthesizer, consists of the following message:

"Warning! Air Defense Warning: Condition Red. Please Acknowledge."

Notice that the visual and auditory alerts carry explicit information about the message requiring the user's attention; often, visual and auditory alerts simply consist of non-specific flashing indicators and auditory "beeps", respectively. We think an explicit auditory message, in particular, is advantageous: the information can be received without requiring a shift in visual attention. Of course, there could be drawbacks to the use of such auditory messages in a battlefield environment, and assessment in an operational context would be necessary before we could unequivocally recommend their use.

User acknowledgement of ADW receipt was accomplished in the following manner. Note that the "RED" ADW is presented in reverse-video in the rectangular shape of a key; the user acknowledges the message by directly touching the ADW status indicator. Once he does so, the "key" is disabled (i.e., the box disappears) and the instruction to acknowledge the message is removed from the reserved message region. Both of these changes are shown in Figure 14.

On many systems, all operator acknowledgments are accomplished by pressing a single key; no unique response is required to differentiate among the different messages that are received. We think our approach, requiring unique, specific responses, which also force the user to look directly at the new information he is acknowledging, is preferable; when a non-specific response is required, we imagined that a user could often issue an acknowledgment without even looking at the message, and would be particularly likely to do so when message load

and the stress of the situation increase. Messages that require an operator's acknowledgment are presumably important; precluding the possibility of his ignoring them reduces the potential for subsequent human error.

3.25 Weapon Control Order

One Weapon Control Order (WCO) message, shown in Table 7, was issued during the scenario. WCOs are accompanied by visual alerts, and, as with an ADW message, user acknowledgment is required.

Table 7
Example of Weapon Control Order Message

1056 (Scenario Time)

FROM: HIMAD
TO: ALL
MESSAGE TYPE: 29 (Weapon Control Order)
MESSAGE:

ORDER NO.	23-12	AREA NO.	2A
PRESENT STATUS		NEW STATUS	
FW	Hold	FW	Tight
DEACT TIME	1210	EFF TIME	1211
RW	Hold	RW	Tight
DEACT TIME	1210	EFF TIME	1211

STATE OF ALERT: Assume Battlestations

For demonstration purposes, we showed the WCO change as though it had just become effective (rather than using the effective time specified in the message, which was outside of the half-hour period of concern). In a fashion similar to the changes effected by the ADW, the receipt of the WCO caused the "FW" and "RW" values displayed in the reserved status area to change from "HOLD" to "TIGHT". The visual prompt to acknowledge receipt of the WCO is displayed in the reserved message area, along with the state of alert message to "Assume Battlestations." This display of WCO information is shown in Figure 16.

Though an auditory warning is not required for WCO messages, we provided one to give a second example of an explicit auditory indication. The auditory alert annunciated by the text-to-speech synthesizer is:

"Note: Change of Weapon Control Status;
"Fixed Wing: From HOLD to TIGHT;
"Rotary Wing: From HOLD to TIGHT;
"Please Acknowledge.
"Alert: Assume Battlestations."

Operator acknowledgment is entered the same way as acknowledgment to an ADW: the new values are displayed, enabled as keys, and the user must touch both the FW and RW values to indicate he has seen them. The change in the displayed information following operator acknowledgment is shown in Figure 16.

3.3 FROM-ABMOC Message Processing

FROM-ABMOC message processing refers to message composition by the ABMOC operator. This function can be activated by selection of the "COMPOSE MESSAGE" function in the Main-menu; the submenu of all messages that can be created by the ABMOC appears in the variable-assignment region of the display screen (see Figure 17). Messages have been grouped into related-topic clusters. By touching the appropriate name, the user would select the message to be created. Only movement order message creation was developed for this demonstration; this is reflected by the presence of a box around "MOVE ORDER" -- signifying that it is an enabled key -- and the absence of boxes around all other message labels. The movement order was selected for implementation because it presented an opportunity to demonstrate a number of interactive display techniques for data entry.

3.31 Movement Order

Samples of various movement orders are presented in Table 8. The message fields and field-values used in the demonstration were based on the particular examples provided in the scenario; methods for entering the data of other message types can be predicted by generalizing from the example we have provided.

Table 8
Examples of Movement Order Message

FROM: ABMOC
TO: FAAR A
MESSAGE TYPE: 13 (Movement Order)
MESSAGE: FAAR A from FH 215354 to FH 116360 at 231800, GS

FROM: ABMOC
TO: B Btry
MESSAGE TYPE: 13 (Movement Order)
MESSAGE: Move weapons to FH 4222 to FH 4422

FROM: ABMOC
TO: E Btry
MESSAGE TYPE: 13 (Movement Order)
MESSAGE: Move weapons to FH 538082

FROM: ABMOC
TO: D Btry
MESSAGE TYPE: 13 (Movement Order)
MESSAGE: Place PLT DS to DISCOM EFF 231200

The sequence of operations for generating a movement order (described below) was determined by analysis of the movement order messages provided. We considered, but did not address a number of issues, including:

- allowing the user to vary the order in which he enters the parameters of a movement order message,

- parameter field "skipping" (i.e., null fields),

- how to provide feedback to alert the user to the fact that he has skipped a field, in case he has done so accidently, or has made an entry that is "tactically questionable" (this would require intelligent software that would not "blindly" report on missing fields or blatant errors, but would parse and assess a message for "meaningfulness"; such assessment would depend, for example, on a knowledge base that describes how assets can move over the given terrain, whether the move is "reasonable" given the current

situation and the location of recent kill reports, whether the move will introduce vulnerabilities in the defense, whether the specified units are "robust" enough -- based on evaluation of supply and equipment reports -- to achieve the specified mission, etc.), and

- revision of a composed, but not yet released message.

Clearly, the definition of an interactive protocol for movement order and other message creation would require that these and other issues be examined.

The first movement order parameter to be specified is the name of the unit who is to move his assets. Following selection of the MOVE ORDER message type, a menu of the potential candidates appears, as shown in Figure 18a. Notice that the menu is specific to the ABMOC echelon. Again, for demonstration purposes as well as to alleviate memory demands placed on the user, an alphanumeric representation (a "trace") of user inputs is displayed at the top of the variable-assignment region; this conversion to the more standard message format is demonstrated in parallel with the data entry functions described below.

The user is prompted for an entry by the green-coded "Move Who" above the menu (following the conventions of this demonstration, it is apparent that only Movement Orders for Batteries (BTRYs) and FAARS were implemented, as indicated by the key-indicator surrounding their labels, and the absence of a box around the other SHORADS C2 units who can be commanded by the ABMOC to change location). Suppose the user selects A BTRY; as shown in Figure 18b, a submenu of A BTRY assets appears below the BTRY column. Had the user selected a FAAR instead, no such submenu would have appeared since such a breakdown is meaningless for sensors.

In this example, the user presses the WPNS key (indicating that we wants A BTRY to move weapons), as shown in Figure 18c; both selections are indicated by the green reverse-video coding. After the user presses the WPNS key, an ENTER key is displayed to the right of the submenu. The availability of this key is suppressed (by not displaying it) until after the user has entered sufficient information for the addressed unit to be specified. When it is possible, inhibiting the availability of keys until they are logically meaningful will prevent inadvertent omission of required data.

Until the user presses the ENTER key, it is possible for him to change his selections for either the primary unit name or the unit's assets.

The next menu of choices to appear allows the user to specify where the designated asset is to move. This "new location" menu is shown in Figure 19a. Two sets of choices were indicated by the movement order examples in the scenario, and are visually clustered on the display.

POINT and LINE allow the user to specify location(s) on the situation display; the listed names allow entry of an AD unit, without specification of the unit's location.

Assume the ABMOC wants the A BTRY weapons moved to cover a line near the FEBA. He presses the LINE key and ENTER key (see Figure 19b), and a new display appears. Figure 20a shows a prompt for selection of a battlefield area. Rather than entering positions as UTM alphanumeric designations, the user will specify a battlefield location in the "language" of the situation display itself by directly touching a location. Because he cannot sufficiently resolve locations on the full-scale map, a blow-up of the desired area is provided in the variable-assignment area. The user touches the general area of the battlefield where he wants the A BTRY weapons to be deployed. As with the INSPECT function (described in Section 3.1 above), a 9 km-by-9 km area is displayed, centered on the location of his touch (see Figure 20b); at the same time, he is given a prompt to "Select first location" (two points are required to define the LINE). The area shown in the high-resolution situation display is indicated on the full-scale map by the green border.

The user touches the high-resolution display to designate the first point, and a green dot, indicating the location of his touch, appears on both the high- and low-resolution displays (see Figure 20c). This location is converted to UTM coordinates (e.g., FH368243) and displayed in the alphanumeric representation of the message. The user can change this first location until he presses the ENTER key. Following entry of the first point, the user is prompted for entry of the second point (see Figure 20d). When a second location is touched, a second green dot and a line between the two points are displayed. Again, the user can change the location of the second point until he presses the ENTER key. (Note: the scale of the high-resolution situation display is sufficient to specify the two-kilometer lines we observed in the data; if it were necessary to specify shorter lines -- i.e., two points with less than a two kilometer separation -- a higher resolution "blow up" would be provided.)

Following designation of the line to which A BTRY is to move weapons, a mission must be specified. This is accomplished with the menu shown in Figure 21, using the same data entry protocol described for all preceding menu selections.

The final entry required is specification of the Date-Time Group (DTG) at which the movement order is to take place. These data are entered using the low-resolution (hour intervals) "digital" clock shown in Figure 22; an analog clock, allowing entries to minute resolution is necessary in an operational situation, and, hence, preferable; one would have been developed had more time been available. Note that the movement is assured to be required sometime during a two-day period, so only day 23 and 24 choices are displayed. Movement order DTG specification at echelons other than ABMOC would require that the

"day-window" be widened (for higher echelons) or narrowed (for lower echelons).

Regardless of which Main-menu function is being used, the operator can always change functions; in composing a movement order, for example, the user might want to move assets that have been inhibited from display during the SET UP DISPLAY operation. The user would select this main-menu function, display the necessary units, return to the COMPOSE MESSAGE operation, and would be returned to the exact point of movement order creation where he had left off to change the situation display.

4.0 Discussion of Alternative Protocols

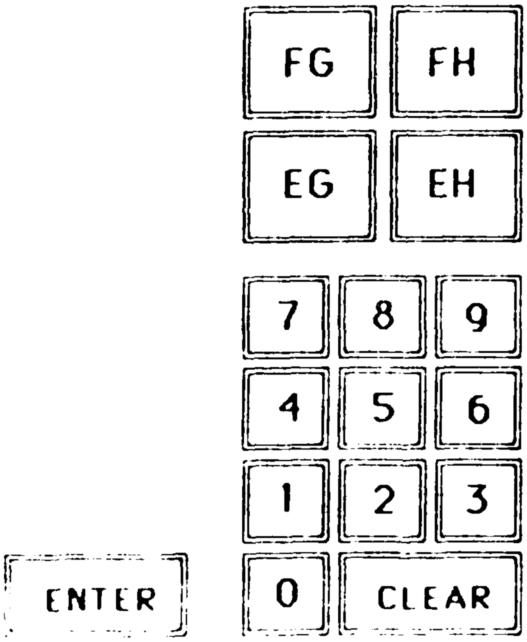
There was insufficient time to demonstrate an alternative protocol for message creation. Our decision to demonstrate an interactive display screen protocol, however, was not arbitrary; in our opinion, the method is preferable, for the tasks performed, to both form-filling and "natural" language using a keyboard. (A more extensive discussion would also cover voice input, used both in "key" replacement and natural language modes.) We briefly present our assumptions and justification for this position.

(1) We assumed that a user would directly select a location for a movement order, as opposed, for example, to simply performing data entry of a specified UTM coordinate. In this latter case, data entry would require a numerical keypad and a special alpha-matrix to designate coordinates for the area of interest. This form of data entry could still be accomplished with a modified data entry keypad displayed directly in the variable-assignment area of the screen; an example of such a data entry "device" is shown in Figure 23.

[Note that the convention of using co-located display and input control surfaces does not imply the use of a touch screen overlay to the display; any cursor positioning device, such as a mouse or trackball, could be used in lieu of the user's finger.]

(2) The device must be distinguished from the protocol used for input. Keyboards may be used for menu-selection, form-filling, and natural language; menu-selection, form-filling, and natural language may all be implemented with voice recognizers. The use of a particular device does not uniquely imply the use of a particular protocol; conversely, the use of a particular protocol does not imply the use of a particular device.

Figure 23
Example of modified data entry "keypad"
for UTM coordinate specification.



(3) The use of a menu-selection protocol implies that there are a limited number of choices for some parameter's value; the user selects one item from a "list" of choices. [A "list" is an enumeration of options, arranged in some spatio-temporal organization; options may be represented graphically, alphanumerically, aurally, etc.] Analysis of the scenario data indicated that, with the exception of location and time specifications (for which none of the above mentioned protocols would suffice), there were very few value choices for each of the parameters of a Movement Order. Whether or not this is true for other messages created by the ABMOC, for all other nodes of the SHORADS C^2 system, or for message communication in any system, would require detailed analysis of possible parameter values for all messages in a system. [We are familiar with Appendix II ("Information exchanged by the SHORAD C^2 system") and are aware that certain parameters of certain messages (e.g., "Air Track Report: AC Type") show long lists of possible values. It may be possible to break down these lists into more logical groupings; further analyses would be necessary to determine if this is operationally beneficial.

(4) To make a meaningful distinction between form-filling and menu-selection, a definition of form-filling must be specified. For example, the Movement Order creation demonstration showed an alphanumeric representation of the data being entered; the entries were placed in fixed fields of an invisible template. Had the template been made visible, the operation would have looked very much like form-filling. But the entry of data was accomplished by menu-selection and "locating" operations.

We restrict the definition of "form-filling" to mean the following: the different fields of a message may be specified, but the possible values for at least some of those fields is extremely large; the structure of the message, then, is fixed and specifiable, but value choices may not be practically enumerated (as is possible with menus). The list of options is generated in, and selected from, the mind of the user. Selection of a value from a (virtual) continuum (e.g., setting a voltage, naming a particular individual in a group of 2000 people) is an instance of form-filling.

(5) Why did we avoid use of a keyboard? First, if the user had been required to fully type out his selections, keyboard entry would have been much slower than the direct selection method. No effective entry (e.g., "A BTRY", "WPNS", "DIRECT SUPPORT", etc.) was one letter in length; but only a single keypress was required to make the entry. In all cases, multiple keypresses would have been required if the selections had been typed out; obviously, such an approach would have taken more time, and introduced errors which would have had to be corrected. The approach we adopted is superior to full keyboard entry.

A second approach would have been to match the number of keyboard strokes to the single button presses required in the direct touch screen selection. This could have been accomplished by labelling (e.g., "1", "2", or "a", "b", etc.) each option displayed on the screen. The user would press the key on the keyboard corresponding to his desired choice. Both data entry methods require a keypress, but the total activity required of the user in the two cases is not identical. A comparison of the sequence of "mental", "perceptual", and "motor" activities required by the two methods is presented in Table 9. This approach to the decomposition of tasks has been described by Card, Moran, and Newell (see Reference [2]). We note that

- the particular breakdown of choice data entry into the subtasks presented in Table 9 is supported by their research; each of the operations listed can be empirically isolated and differentiated from the others, and

- there is overwhelming evidence from their research (and from the experimental literature, in general) that each of the subtasks enumerated take real time (varying from a few hundred milliseconds to a second or more).

Table 9
Comparison of Keyboard and Interactive Screen Methods
for Data Entry of Parameter Choice

Action	Keyboard	Interactive Screen
Find Choice (read options)	X	X
Read Label of Desired Choice	X	
Move Eyes to Keyboard	X	
Find Key Matching Label	X	
Finger to Key	X	X
Press Key	X	X
Return Eyes to Display	X	

Most military personnel are poor typists; it would take them longer to do keyboard data entry than interactive screen data entry. (Whether or not it would take longer for skilled typists is an empirical issue.) Since the selections and data entry to be made by the user do not require a keyboard (i.e., all tasks can be accomodated by the interactive screen method), there is no justification for its use. In

fact, the prediction of longer data entry times are an argument against its use.

Note, by the way, that the superiority of the interactive screen method does not depend on the use of a touch screen overlay, which, as others have quite legitimately argued, can result in fatigue of the user's arm if used for extended periods of time. A mouse, which does not produce such fatigue, could have been used for direct screen interaction for selecting options or designating locations. The critical difference between interactive screen and keyboard entry techniques is that the latter requires the user to look away from the display screen while the former does not. It is the time lost in shifting visual attention, and in performing cognitive, perceptual, and motor reorientation to different display/control devices that argue against the use of a keyboard when the tasks to be performed do not require one.

5.0 Future Studies

We suggest that future studies on interactive protocols address the requirements of a particular application (i.e., a system, like SHORADS C^2). That is, a detailed functional specification of the user tasks must be available to determine which kinds of protocols and input/output devices should be considered. In the preceding section, our justification of an interacti[illegible] display, menu-selection, and locating approach depended on analyses of the specific data that were made available. If, for example, the SHORADS C^2 system had a requirement for free-format message construction, some level of "natural" language processing and use of a keyboard and/or voice recognizer would be required; analysis of a natural language protocol would be meaningful in such a context.

Second, the experimental context in which protocols are evaluated will determine the generalizability of the results to an operational context. The results of an experiment in which many subjects are sampled for short periods of time will be quite different from a sample of a few subjects who are run for extended periods of time. The latter methodology is appropriate if it is on-going operational performance, rather than novice behavior, that is to be predicted.

The collection and analysis of data from such an operational context would require:

- on-line (i.e., computer-based) data collection and analysis programs, and

- an operational testbed which simulates the activities of the operational situation with a high degree of versimilitude.

We note that on-line data collection and analysis routines would have more general applicability: they could also be used as one component of embedded training. They would also be used during ongoing operations to provide a quantitative basis for determining when feedback should be given to a user. (A more extensive discussion of the requirements of in-line "human performance monitoring" may be found in References [3] and [4].)

An operational testbed could be effectively constructed from nodes residing at different geographic locations. Such nodes would be linked by a commercial network; data transmission rates would be no worse than those currently anticipated for military systems. Such a distributed system would include the following advantages for the design, development, and evaluation of experimental systems:

(1) no single laboratory would bear the burden of constructing the entire system, yet would enjoy the benefits of evaluating a node within a larger system context,

(2) demonstration of the system would be possible at a number of locations (thus reducing travel requirements),

(3) development work at different sites would be available to all researchers on an almost instantaneous basis, which would effectively extend the research community of all groups involved; sharing of resources (e.g., technical expertise, software capabilities, etc.) and critical interchanges would be a continual process, rather than events that take place only occasionally at technical meetings, and

(4) it is representative of the real situation in an operational environment.

Our last comment is concerned with the tools that are available for prototyping and evaluating human-machine interface designs. Too often the majority of design and evaluation time is spent on tasks that are irrelevant to the issues at hand: hardware interfacing problems, device drivers, software development of screen formats, etc. Consequently, the empirical evaluation of design alternatives is cursory, at best, and design errors become apparent when it is extremely difficult, time-consuming, and expensive to correct them: that is, after the system has been developed and fielded.

The development of the extensive demonstration described in this report was only possible through the use of a very high-level interactive display prototyping language; most of our time was spent on analyses and thinking about the interaction between the computer and the human, not on software development or hardware concerns. The demonstration development took about 120 hours; we think we should be able to prototype systems even faster (e.g., reduce the 120 hours we spent to about 40).

Such rapid development is only possible if "rapid prototyping" tools are available. We are beginning a number efforts to develop such tools. One tool would accelerate the development of console "data", such as the interactive display screens described in Section 3.0. Essentially, time would be saved by allowing the developer to "program" directly in interactive graphic terms -- e.g., draw things directly on the display screen -- rather than having to actually write software code. This tool (an extension and enhancement of Easel, the interactive graphics language currently in use in our laboratory) would be developed in a way that gave it both device independence and portability to different operating systems (e.g., RSX, VMS, UNIX). The end result would be that, regardless of devices or computing environment, a single high-level development "language" could be used to develop human-machine transactions. Since this development would be so swift, alternative protocols could be prototyped and empirically evaluated within short periods of time.

Our other "rapid prototyping" activities address the data development requirements of the Generalized Human-Machine Interface, a system under development at Lockheed for the past four years. A description of these tools can be found in two proprietary reports (see References [5] and [6]).

6.0 References

[1] MIS 34585: "SHORAD C^2 System Specification" (10 October 1983).

[2] Card, S. K., Moran, T. P., and Newell, A. The Psychology of Human-Computer Interaction. Lawrence Erlbaum Associates, Hillside, New Jersey, 1983.

[3] Knox, R. E. "Development of a Generalized Human-Machine Interface. In Proc. 6th MIT/ONR Workshop on Command and Control (Boston, 25-29 July 1983), pp. 100-105

[4] "Development of an Generalized Integrated Human-Machine Interface." Lockheed Proprietary IR&D Report for 1982-1983. March, 1983.

[5] "Development of a Generalized Human-Machine Interface." Lockheed Proprietary IR&D Report for 1984-1985. March, 1985.

[6] "Artificial Intelligence / Expert Systems for C^3I." Lockheed Proprietary IR&D Report for 1984-1985. March, 1985.

7.0 Appendices

Appendix 7.1

Summary of HEL/AAAD and LEC 1-day meeting: 7 December 1984

Meeting Date:	7 December 1984
Location:	Lockheed Electronics Company, Plainfield, New Jersey
Purpose:	Show Interactive Protocols demonstration and hold discussions

Attendees:	HEL/AAAD	Lockheed
	Sgt. Jim Bevins	Mr. Joseph Barletta
	Dr. Jon Fallesen	Mr. Richard Brandau
	Major Marcus Cox	Mr. James Coburn
	Major Roger Duckworth	Mr. Paul Corrigan
	Major Roger Parks	Mr. William Guyton
	Mr. Christopher Smyth	Ms. Merryll Herman
		Mr. Jules Kaplan
		Dr. Rita Knox
		Mr. Don Rowan
		Dr. Eric Sigman

Certain revisions were requested during the morning demonstration session. The revisions were the changes in symbology coding described in Section 3.1 above. These revisions were incorporated and demonstrated in the afternoon. Such rapid changes are made possible by the high-level development tools used in the Human Machine Interface program.

During the afternoon session the following issues were discussed:

o Symbology — Should users be able to specify their own, personalized symbol sets? Should one standard symbology set be used, or are multiple sets (differing in complexity as a function of the operation in which they are used) required?

o Primer — Embedded training is a requirement of new systems under development. Operational systems will be required to have extensive help facilities.

o Data Display — When should different data be displayed? How and when will data be selectively filtered from displays as a function of the current activities of a user?

When should displayed data be removed from the display? Should there be built in "time-out"

Appendix 7.2

"Interactive Protocols for generalized human-machine interface"

functions? Should the displayed data be replaced when updates of the same type are received? Should the user be able to specify that he wants data to be removed? How does he accomplish this?

- Protocols — What are the candidate alternative protocols? Can they be compared through analysis? Are demonstrations necessary? What quantitative data should be collected? What is the appropriate context for the collection of the data (can generalizations be made from data collected during half-hour experimental sessions, or must data be collected in an operational context)?

- Input Devices — Does the separation of display and control surfaces (such as is found when information is displayed on a screen and inputs are entered from a keyboard) impair performance? What should the input device be (mouse, joystick, touch screen overlay, etc.) when display and control surfaces are co-located?

- Who's the User? — Will the Officer-in-Charge (OIC) ever want to directly control communications (e.g., directly interact with the display screen) or will communications always be mediated by a subordinate (an operator who is making data entries based on the OIC's directives)?

  What is the appropriate user profile? What time frame is being considered for implementation of new systems? Will fewer individuals, each having higher average capabilities than the current average military personnel, be using these systems?

  Will there be more than one user of each system? Will individual use be task-dependent and vary as a scenario develops? Should the system be adaptable and modifiable as a function of who the user is?

To appear in Proceedings of 7th MIT/ONR Workshop on C^3 Systems (11-15 June 1984, San Diego, Cal.)
Publication date: December 1984.

INTERACTIVE PROTOCOLS FOR GENERALIZED HUMAN-MACHINE INTERFACE

Rita E. Knox and Richard Brandau

Lockheed Electronics Company, Inc.
1501 U.S. Highway 22, CS #1
Plainfield, New Jersey 07061

Summary

Communication requirements to be supported by the Generalized Human-Machine Interface (GHMI) fall into two general categories: communication intelligence, which includes assumptions made by communicators, topical context, style, and versatility of information representations, and the representations themselves, which are the information-to-data and data-to-information translation rules. This paper discusses the representation problems of human-machine communication within the context of more general features of human communication.

Introduction

Interactive communication between human and machine (i.e., computer) has, for the most part, been mediated by keyboard and display screen: the user issues a statement which must obey the rules of some well-defined command language and the machine answers by sending a statement to the screen and/or by changing some internal state.

An alternative approach, which allows humans to communicate with computer systems in much the same way as they communicate with each other, is adopted by the Generalized Human-Machine Interface program (Knox [9]). Some people have argued that an inter-human model may be inappropriate for human-machine communication. They point out, for example, that there is no inter-human exchange that is analogous to certain graphics commands, or the manipulations of graphic symbols on a display which those commands achieve.

First, our use of "human-human communication" as a model for human-machine communication is overly restrictive. Humans interact with each other and with inanimate objects; features of manipulating inanimate objects must be captured in human-machine communication. The concept of rotating an object on a screen is not foreign to human beings who have rotated real objects in space thousands (millions?) of times during their lifetimes. The appearance of rotating objects in the two cases is different, as are the specific actions taken to accomplish rotation in the two worlds. Central issues of the representation problem are (1) how different the visual features of the two events can be and still retain a correspondence between the real and artificial worlds be apparent and (2) what should be the relationship between motor actions to move real objects and motor actions to move artificial objects.

Second, though there are computer-mediated actions for which there are no real-world analogues (e.g., manipulations that violate the laws of physics), does this mean that the rules of real world interactions are irrelevant to human-computer interaction?

Third, there are features of inter-human communication, such as our ability to follow transitions between topics, which are not generally available in human-machine communication (Bannon, et al. [1]); their absence limits human exploitation of computer capabilities. These features, such as human attentional abilities, are overlooked when criticisms of the inter-human model of human-machine interaction arise.

Increasing emphasis is being placed on the importance of natural language processing for human-machine communication (see Strategic Computing [14]). While we agree that natural language is an important capability, it is only one vehicle of inter-human communication. If and when full natural language processing by machines is achieved, it will not suffice as the medium of expression for all ideas we might want to exchange with machines.

When people communicate, they select and use (albeit, not necessarily with any awareness) an array of communication tools that are appropriate for the information being communicated and useable under the constraints of the communication environment. We talk, listen, write, draw, gesture, point, add vocal emphasis, and so on, as the conditions of the communication event warrant and allow. That is, there is a relationship between the information to be communicated and the ways it can be represented; moreover, the circumstances governing the communication may impose limits on the variety of representations that will prove useful. For example, drawing and gesturing are useful in direct communication, but are ineffective during a phone conversation.

To employ the tools of human communication in human-machine communication we must understand how humans map task-relevant information into data representations, such as sentences, pictures, and gestures. Although we are primarily concerned here with the representation problem, we cannot deal with it independently of other human information processing capabilities.

Many researchers are studying the problem of human-machine interface (HMI, and, variously, "man-machine interface," "user-computer interface," etc.). We note here that in most cases the domain of inquiry is narrower than ours in that the definition of human-machine interface extends only to the data representation problem. Since others have not addressed the problem within the context we have used for the development of the GHMI, the correspondence between their definition of protocol features and ours is not clear cut. We will attempt, however, to describe the relationship between our work and others where appropriate.

Human Communication Capabilities

Communication between humans involves more than the exchanging of ideas. Representation of ideas is only one step in a process that includes:

- searching memory for information which is relevant to the current task of the current topic,

- sometimes discovering that information is missing,

- selecting (or dynamically building - see Chafe [5]) the idea-path through the larger store of memory-resident information (i.e., semantic memory, episodic memory - see Lindsay and Norman [10], Kintsch [9], respectively) which will be expressed to communicate information, or a request for information,

- organizing the sequence in which the ideas are to be expressed, and finally

- codifying the ideas in a form which can be communicated, and then externalizing them (i.e., response generation).

An exhaustive listing of human representation phenomena is not possible here. However, there are certain salient capabilities that have been considered in the design of the GHMI.

- the same (or similar) meaning is captured by different physical representations. Consider, for example, the various external forms that can be aroused by the concept of a "rose."

- While we have the ability to extract a common concept from different external representations, we do not discard the modality- or representation-specific information about how the concept was communicated. That is, we are capable of remembering the particulars of how we learned a particular piece of information. If asked how you knew how to get someplace, you would probably remember if you had been given instructions on the phone or if someone had drawn you a map.

- Formally equivalent representations do not necessarily convey the same meaning, as many of us experience when looking at an equation as opposed to looking at a graph of the same equation.

- Most objects (or events) have multi-modal attributes. A rose has color, form, scent, smooth and sharp textures; a moving automobile has correlated auditory and visual features which change over time.

If we are to incorporate these human representation capabilities in a human-machine interface our design must meet (at least) the following requirements:

- The concept of different representations for the same information must be present. A sample of possible alternative representations is presented in Figure 1. The interface must be capable of generating alternative representations for output to the human, and of reducing different, but equivalent data inputs to the same internal information. Ideally, the one-to-many and many-to-one mappings would be specified algorithmically. At least for the present, however, the various representations will be defined on a case-by-case basis since a general algorithmic description requires an understanding of human representation capabilities which is currently unavailable. (In fact, it is an area that is in need of research.)

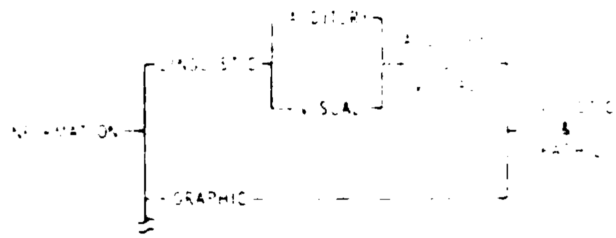


Figure 1

- Multimodal attributes of a particular event should be captured. For example, it should be possible for someone to say "Move this one" and have the machine understand that the object one is pointing at is the referent of "this" (see Schmandt and Hulteen, [12]).

- The interface must determine which of the representations for some information should be used at any particular time. We don't think the choices are arbitrary, but depend on a number of things. First, while more than one representation can be used, a particular representation may be the more natural choice for certain information. Moreover, the representation selected may depend on the particular user; so the system must know what the relevant user attributes are in determining the use of a particular representation. Second, input/output (i.e., human/computer) representations during a particular task should be "matched." For example, while it might be possible to communicate certain information either by talking or by writing, it would be a peculiar conversation in which one party talked and the other party wrote.

Relevant GHMI Components

The GHMI system under development supports human-machine communication by performing natural language processing, attention monitoring to provide context and sequencing to user inputs (and eventually host or external inputs), dynamic device assignment to reconfigure the use of console devices in near real-time, and human performance monitoring to detect, categorize, analyze, and provide feedback on user errors. The translation between device-specific data and internal GHMI representations is performed by decode and encode modules of the system. A data management structure (see Brandau and Fox [3]) allows, among other things, the entry of user-defined commands to off-load data monitoring and mental calculations to the machine. All application-specific data used by the GHMI system is represented entirely as data; this allows new applications to be developed through the authoring of data tables, and without modification of processing code. A diagram of the system architecture appears in Figure 2.

In the GHMI, protocol and representations are governed by the Decode module (of which Natural Language Processing is a part), the Encode module, the Attention Monitor, and the Dynamic Device Assignment module.

Decode accepts an input from the user, strips away device-specific information, converts the statement to a standard internal format, and passes the command to the Attention Monitor. This module determines which topic the "object" of the command belongs to. A

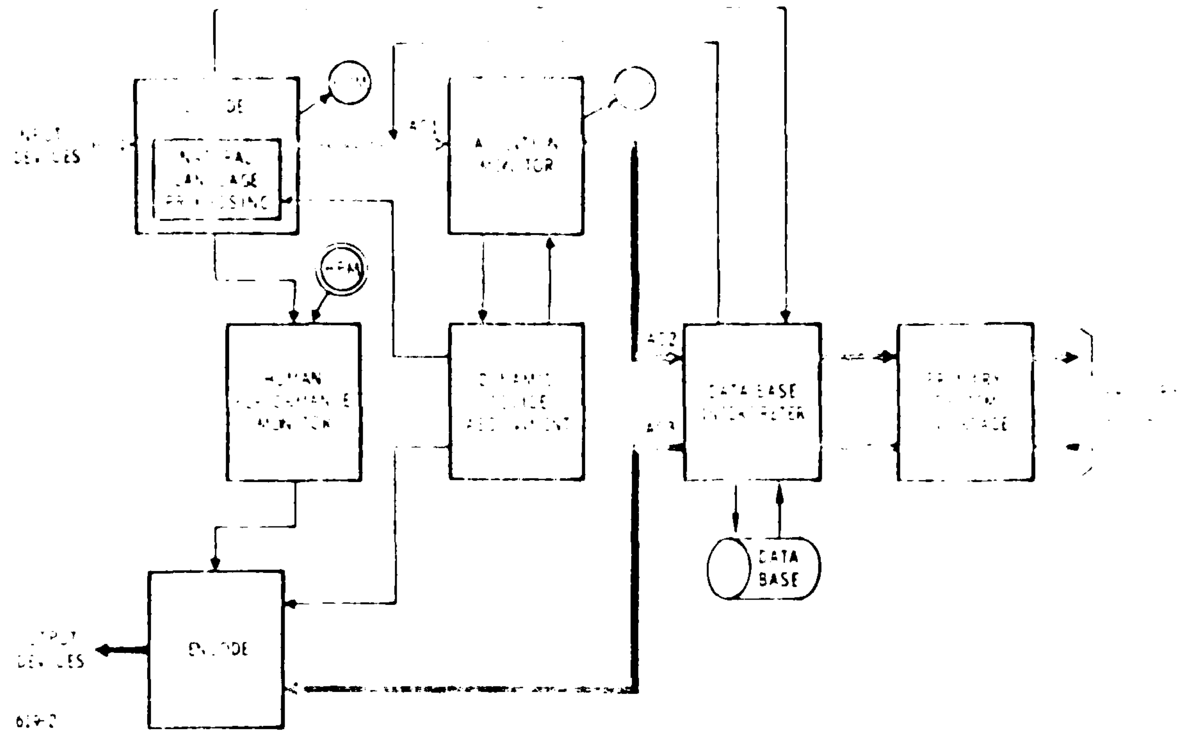


Figure 2

'topic' is defined as a collection of tasks that can be performed on objects. Topic identification requires that things are talked about in a particular sequence. (The sequential constraints that are active at any time are path-dependent and, may, in fact, specify that any number of objects may be talked about in any order.) If the object of the command is part of an active topic, the topic identifier is appended to the command before it is passed to the Data Base Interpreter for execution. If the object is part of a new or deferred topic, continued conversation will only be possible if the topic can be (re)activated. This depends on a determination by the Dynamic Device Assignment module as to whether or not virtual devices (e.g., different quadrants of a display) required by the topic can be made available. Briefly, the decision depends on the currently active topics, their priorities, the virtual devices in use, alternative virtual devices that may be used by each of the topics, and the "disruption" which may be caused by reassigning active topics to new virtual devices, and, sometimes, new representations. The crucial point here is that the activation of a topic and the representation of the information of that topic depend on the order in which preceding topics have been activated, and on the particular array of topics to which a new topic may be added, as well as on inherent properties of the topic itself.

If the topic succeeds in being activated, the virtual device assignment information and activation 'trigger' are passed to the Encode module. All subsequent transactions on this topic are mediated by the particular data representations defined and processed by the Decode and Encode modules.

Interactive Protocol Issues

In this section we describe a few human-machine representation features that have been proposed by others, and discuss their merits in terms of the assumptions made in the design of the GHMI.

A major advance in human-machine communication was the development of the multiple window concept; it allows the material for a number of user-activated tasks to be simultaneously presented on a display. The dimensions of windows may be user-defined and the windows themselves may be "stacked." While the ability to simultaneously display and access multiple tasks is a feature of the GHMI, the overlapping of task-windows is deliberately prevented for a number of reasons. First, the GHMI is a mediator between two parties (as shown in Figure 3-a). Traditional "windowing" gives control of the display, and, hence, the topics of conversation, to one party of the conversational dyad: the human sitting at the console. If the human and machine (or human and human if the "machine" is a computer-mediated network of humans) are engaged in work where they have complementary assignments, there could be serious consequences if the machine's priorities are not given consideration in the configuration of topics on the display. Consider, for example, the situation illustrated in Figure 3-b. P(articipant)1's topic priority ordering is a-b-c-d; P2's is c-b-a-d. If windows are arranged according to P1's preferences, P2 will not be able to communicate about its highest priority topic - c; if performance of a system depends on both participant's work, we would expect it to be seriously impaired by P1's ignorance of P2's information. (In other words, P2 is actively working on topic-c and the data on the topic-c window is changing; however, if the windows are arranged to P1's ordering he will be unaware of P2's [illegible].)

Second, even if the GHMI is mediating between a human and a system "slaved" to that human's commands, the traditional windowing approach puts the burden on the human to remember where topic-pages are located (just as he or she must search for or [illegible] an important slip of paper has been [illegible] a stack). Third, the work to rearrange the topic-pages to bring those of current interest to the "top" ("reconfiguration") must be performed by the user.

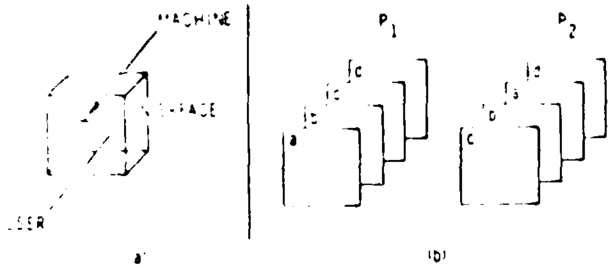


Figure 3

Only non-overlapping windows are allowed in the GHMI system — nothing precludes the use of overlapping windows; they simply are not defined in the Dynamic Device Assignment data tables). The work of detecting requests for or inferring the currently pertinent topics, presenting the corresponding display-control windows, and activating/suspending topics as necessary is carried out by the Attention Monitor and Dynamic Device Assignment modules. We have anticipated the requirement but not yet implemented a facility that would allow the human to override those modules' decisions; after all, since humans can make errors in judging another's dynamic priorities, the GHMI can be expected to make mistakes, too. However, the benefits of having the computer take care of the "administrative" details of keeping information at the user's fingertips outweigh the occasional delays that will be introduced when the user must repeat (or explicitly state) that a particular topic is required immediately.

The most commonly used forms of human-computer protocols have been menu-selection, form-filling, command languages, and, more recently, limited "natural" language. Clearly, this is not a complete list of options. For example, specifying a point on a display screen is neither form-filling nor natural language, although graphic manipulations may be similar to natural language in certain formal respects. We would also argue that it is not menu-selection, although it could be formalized as a "pick 1-of-n pixels" operation. However, even on very low resolution displays it is doubtful that the human conceptualizes the problem as a menu-selection operation.

Even in the case of an input being clearly categorizable into one of the four classes, all inputs of the same type do not involve identical human operations. For example, the following are all examples of menu-selection:

- selecting 1-of-n graphic symbols,
- selecting 1-of-n alphanumeric options,
- selecting 1-of-n options when multiple pages are required to display all n options, and
- selecting 1-of-n options from a menu where the available options are determined by a previous selection;

However, the problem domains, the cognitive processes, and the response procedures involved in the four cases are different. Should the menu formats differ?

The nature of the information to be communicated places constraints on the protocols that can be used; for example, the response set for an entry may be too large to make menu presentation feasible. The way in which data are displayed may imply what protocol options may be used and may even preclude certain options. In other words, the protocol is bi-directional and is not simply determined by how the user may enter data to the computer, but must also consider how the computer presents data to the human. Even when more than one protocol is logically feasible, it may be desirable (e.g., to minimize errors) to allow only one protocol type to be used; for example, to require the explicit response to be selected from a well-defined response set.

Even as the set of different protocols for human-machine interactions grows, some basic problems remain to be studied:

- What is the relationship between the data required from the human or machine and the protocol to be used?

- What is the relationship between a display format, for example, and the implied protocol to be used?

- What are the behavioral and operational conditions which suggest when there should be many or few protocols for the user to choose from?

Interactive Languages

There has been much recent work on the use of formal grammars as the specification language for the protocols of human-machine interface (Reisner [12]; Jacob [7]; Bleser and Foley [2]), and discussions of the relative merit of different, but formally equivalent representations (Jacob [6]).

Reisner has argued for and demonstrated the predictive importance of specifying the cognitive operations (e.g., memory search) along with the observable actions (e.g., button presses) required to perform some function. This modelling of cognitive operations in the analysis of human-computer interaction has been developed more fully and described by Card, et al. [4]; their "Model Human Processor" is defined by parameters of human information processing which have been discovered through basic research. These models are extremely useful in assessing alternative protocol designs for well defined tasks such as text-editing; however, they do not capture many human capabilities described in the preceding sections.

Formal descriptions of the 'user-computer interface' have included the specification of semantic, syntactic, and lexical levels.

The semantic level specifies the actions to be executed by the computer upon receipt of particular commands, independent of the form in which the commands were issued by the user.

The syntactic level describes sequential constraints on "words" (or tokens) which, when satisfied, uniquely define a system action. The idea is that a "word" can have more than one interpretation based on the context in which it appears; moreover, a sequence of words (a "sentence") may have an interpretation based on sequences that have preceded it.

The lexical level describes the device-specific realization of the "words" at the interface. It includes the description of graphic elements, their position on a display screen, their color, etc.; the manipulation of input devices (e.g., mouse, keyboard) to form input tokens; the vocabulary of a command language.

While we understand this approach to formalization, these grammars do not capture all features of human communication which GHMI attempts to imitate.

A formal specification within the context of GHMI must capture all available representations for the information of a particular task. This, in itself, does not present serious problems for these formal grammars. The difficulty arises in specifying the conditions for the selection of a particular protocol with a particular representation. The state upon which the selection depends must include other active tasks and the representations (hence, the virtual devices) they use. This configuration of active tasks depends, in turn, on the sequence in which the tasks were activated, their priorities, the array of representations available to each of the tasks, and conflicts between tasks for the particular devices required for their data representations. At some point, the state specification will include a description of the user; this description will not only reflect static attributes (e.g., the traditional "novice" and "expert"), but will provide a dynamic assessment of a user's performance at a particular time on a particular day. These states are considerably more complex than those described by previous research. At this point we do not know if formal grammars are appropriate predictive, analytic, or descriptive tools for representing the GHMI system.

<u>Representations at the Interface</u>

In this section we briefly describe some of the problems to be solved in the explicit representation of information. We will restrict this discussion to interactive display screens; a more general treatment would include such things as keyboard and voice entry and voice response. We will not be concerned here with specific device technologies.

It has been argued that software programmable controls should both look and be operable like the hard-control counterparts they are replacing (Nakatani and Rohrlich [11]). The limitation to this approach is that there are many new processes that have no hard-control predecessors; it is not clear that an interactive language designed to accommodate all dialogues within a new system will allow the incorporation of the hard-control analogues.

The data on an interactive display screen (i.e., one that mediates both display and control functions) must not only convey the information of a topic, but must also communicate protocol rules and the current state of the system.

For example, in the GHMI we use the convention that rectangular labelled keys mediate on-off (more generally, binary state) functions, and this coding is used consistently (see Bleser and Foley [2]; Nakatani and Rohrlich [11]). We originally coded the off-state as a dashed line, the on-state as a solid line (see Figure 4-a). As it turned out, the two states were not sufficiently distinctive. We have since changed the state indication to that shown in Figure 4-b: the on-state is diffentiated from the off-state by the addition of a solid bar across the top of the key. So, the on-off function is implied by the shape of the key, and its state is indicated by the presence or absence of a bar.

The binary-state key example is fairly simple. Consider the functional properties of the operation illustrated in Figure 5. Here we have a data calculation task where a user may request the distance between an object (e.g., a target or land mass) and his "own ship" (operator "f") and/or the projected position of the object in five minutes (operator "g"). The user must specify both the operator and the operand for the calculation to be executed. Among the alternative syntaxes that could be followed are:

(1) The user must select one operator followed by one operand and the sentence is complete (a).

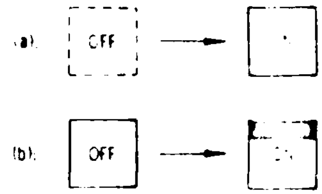
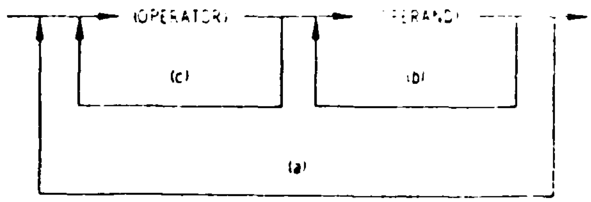


Figure 4



(a) f(x), g(x)

(b) f(x), f(y), f(z)

(c) f(g(x))

f. COMPUTE x's DISTANCE FROM OWNSHIP

g. PROJECT x's POSITION IN FIVE MINUTES

Figure 5

(2) The user must select one operator followed by one or more operands. The sentence is complete when operand selection is ended (b).

(3) The user may select one or more operators before selecting a single operand. The interpretation of successive, contiguous selection of operators is a composition function (c).

(4) The user may select one or more operators (as in 3) followed by one or more operands (as in 2). The interpretation is a composition function applied to multiple objects (case not shown).

The calculated results of the four syntaxes will be different (provided, of course, that multiple operators and/or operands are chosen when the option is available). Assume they are all meaningful to some system. Do we design the task and the display screen to allow any of the four syntactical structures to be used (as opposed to designing four separate tasks, each corresponding to one of the syntaxes)? How are the syntactical constraints made apparent in the visual features of the interactive display?

Suppose we have designed a task using the simplest syntax: one operator followed by one operand (1). In the example shown in Figure 6-a, "HELP" and "READ-OUT" are operators, "A," "B," "C," and "D" are operands. Operators are coded as circles. The user decides to ask for help (i.e., "picks" the HELP key); this moves the task into the state where an operator is active and an operand must be selected.

In Figure 6-b, the following state information is present:

- "HELP" is active (indicated by the filled square to the lower right of the key).

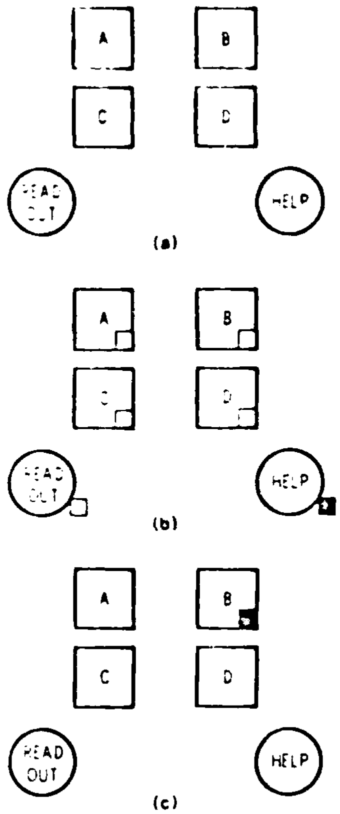


Figure 6

- Legal operands of the HELP operator are A, B, C, D, and READ-OUT as indicated by the open square in the lower right of the respective keys.

The user chooses B as the operand. As shown in Figure 6-c, the small square within the B key is filled and the squares to the lower right of other keys are removed. Although not shown here, the help information would be displayed. The positioning of the filled square in the lower right has significance: it "labels" the displayed information as help information (rather than read-out information) by using the following visual analogy:

Position of : B Key :: Position of : Task Frame.
Square in Help Key
B Key

The reader can imagine far more complicated operations than the one described here. The point to be made is that the design of the interactive syntaxes and of display screens which support them can be terribly complex. Something like the formal grammars described in the preceding section will be required to describe them. If we are right, though, in requiring state dependencies of the sort we have described, much more powerful analytic design tools must be developed.

Conclusion

The purpose of this paper has been to describe some of the problems that will require solutions if human-machine communication is to capture the richness of inter-human communication. While we know that the concepts introduced so far in the development of the Generalized Human-Machine Interface are only the rudiments of what will be required, we are certain that our approach offers greater potential than those which address the representation of information without consideration of the intelligence that interprets it.

References

[1] Bannon, L., Cypher, A., Greenspan, S., and Monty, M.L. Evaluation and analysis of users' activity organization. In Proc CHI'83 Human Factors in Computing Systems (Boston, 12-15 December 1983), ACM, New York, pp. 54-57.

[2] Bleser, T. and Foley, J.D. Towards specifying and evaluating the human factors of user-computer interfaces. In Proc CHI'82 Human Factors in Computing Systems (Gaithersburg, 15-17 March 1982), pp. 309-314.

[3] Brandau, R. and Knox, R.E. "Current": A dynamic donor-recipient object base for tactical human-machine transactions. In Proc. 7th MIT/ONR Workshop on C*3 Systems (San Diego, 11-15 June 1984).

[4] Card, S.K., Moran, T.P., and Newell, A. The Psychology of Human-Computer Interaction, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1983.

[5] Chafe, W.L. Creativity in verbalization and its implications for the nature of stored knowledge. In R.O. Freedle (Ed.) Discourse Production and Comprehension Volume I. New Jersey: Ablex Publishing Corporation, 1977, pp. 41-55.

[6] Jacob, R.J.K. Using formal specifications in the design of a human-computer interface. In Proc CHI'82 Human Factors in Computing Systems (Gaithersburg, 15-17 March 1982), pp. 315-321.

[7] Jacob, R.J.K. Executable specifications for a human-computer interface. In Proc. CHI'83 Human Factors in Computing Systems (Boston, 12-15 December 1983), ACM, New York, pp. 28-34.

[8] Kintsch, W. The Representation of Meaning in Memory. New Jersey: Lawrence Erlbaum Associates, 1974.

[9] Knox, R.E. Development of a generalized human-machine interface. In Proc. 6th MIT/ONR Workshop on Command and Control (Boston, 25-29 July 1983).

[10] Lindsay, P.H, and Norman, D.A. Human Information Processing: An Introduction to Psychology (2nd Edition). New York: Academic Press, 1977.

[11] Nakatani, L.H., and Rohrlich, J.A. Soft machines: A philosophy of user-computer interface design. In Proc CHI'83 Human Factors in Computing Systems (Boston, 12-15 December 1983), ACM, New York, pp. 19-23.

[12] Reisner, P. Further developments toward using formal grammar as a design tool. In Proc CHI'82 Human Factors in Computing Systems (Gaithersburg, 15-17 March 1982), pp. 304-308.

[13] Schmandt, C. and Hulteen, E.A. The intelligent voice-interactive interface. In Proc. CHI'82 Human Factors in Computing Systems (Gaithersburg, 15-17 March 1982.), pp. 363-366.

[14] Strategic Computing. New Generation Computing Technology: A Strategic Plan for its Development and Application to Critical Problems in Defense. Defense Advanced Research Projects Agency. 28 October 1983.

Appendix 7.3

Demonstration Photographs

END
Feb.
1988
DTIC